

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/005136

International filing date: 22 March 2005 (22.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-107778  
Filing date: 31 March 2004 (31.03.2004)

Date of receipt at the International Bureau: 28 April 2005 (28.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2004年 3月31日

出願番号  
Application Number: 特願2004-107778

パリ条約による外国への出願に用いる優先権の主張の基礎となる出願の国コードと出願番号  
The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is

出願人  
Applicant(s): 松下電器産業株式会社

2005年 4月13日

特許庁長官  
Commissioner,  
Japan Patent Office

小川



【書類名】 特許願  
【整理番号】 2048160031  
【提出日】 平成16年 3月31日  
【あて先】 特許庁長官 殿  
【国際特許分類】 G09C 5/00  
【発明者】  
【住所又は居所】 大阪府門真市大字門真1006番地  
【氏名】 布田 裕一  
【発明者】  
【住所又は居所】 大阪府門真市大字門真1006番地  
【氏名】 山道 将人  
【発明者】  
【住所又は居所】 大阪府門真市大字門真1006番地  
【氏名】 大森 基司  
【発明者】  
【住所又は居所】 宮城県仙台市青葉区上杉5-8-7-606  
【氏名】 静谷 啓樹  
【発明者】  
【住所又は居所】 宮城県仙台市青葉区星陵町3-36-203  
【氏名】 満保 雅浩  
【特許出願人】  
【識別番号】 000005821  
【氏名又は名称】 松下電器産業株式会社  
【代理人】  
【識別番号】 100090446  
【弁理士】  
【氏名又は名称】 中島 司朗  
【手数料の表示】  
【予納台帳番号】 014823  
【納付金額】 21,000円  
【提出物件の目録】  
【物件名】 特許請求の範囲 1  
【物件名】 明細書 1  
【物件名】 図面 1  
【物件名】 要約書 1  
【包括委任状番号】 9003742

## 【書類名】特許請求の範囲

### 【請求項 1】

群 G 用いて、2 個以上の整数の加算を行う加算プログラムであって、

前記群 G は、真に含む部分群 S をもち、

前記群 G 上の幂演算を行うことにより、前記整数を前記 G に属する元に変換する変換モジュールと、

前記群 G 上の基本演算を行う主要演算モジュールと、

前記群 G または、前記群 G の部分群 S における前記変換モジュールで行う変換の逆変換を行う逆変換モジュールと、を備えることを特徴とする、加算プログラム。

### 【請求項 2】

前記群 G は、剩余整数環の乗法群であることを特徴とする、請求項 1 記載の加算プログラム。

### 【請求項 3】

前記群 G は、複数個の相異なる素数  $p_1, p_2, \dots, p_k$  ( $k > 1$ ) の積  $n = p_1 \times p_2 \times \dots \times p_k$  に対し、 $\mathbb{Z}/n\mathbb{Z}$  の乗法群であることを特徴とする、請求項 2 記載の加算プログラム（ただし、 $\times$  は乗算である。また、 $\mathbb{Z}$  は整数環であり、 $\mathbb{Z}/n\mathbb{Z}$  は整数を  $m \bmod n$  した値で構成される剩余整数環である）。

### 【請求項 4】

前記逆変換モジュールは、前記素数  $p_1, p_2, \dots, p_k$  を用いた  $\mathbb{Z}/p_1\mathbb{Z}, \mathbb{Z}/p_2\mathbb{Z}, \dots, \mathbb{Z}/p_k\mathbb{Z}$  の乗法群における離散対数問題を解くことを特徴とする、請求項 3 記載の加算プログラム。

### 【請求項 5】

前記逆変換モジュールは、前記素数  $p_1, p_2, \dots, p_k$  を用いた  $\mathbb{Z}/p_1\mathbb{Z}, \mathbb{Z}/p_2\mathbb{Z}, \dots, \mathbb{Z}/p_k\mathbb{Z}$  の乗法群における離散対数問題の解に対し、中国人の剩余定理を用いることを特徴とする、請求項 4 記載の加算プログラム。

### 【請求項 6】

前記群 G は、2 つの素数  $p, q$  と正整数  $m$  を用いて表される  $n = p^m \times q$  に対し、 $\mathbb{Z}/n\mathbb{Z}$  の乗法群であることを特徴とする、請求項 1 記載の加算プログラム（ただし、 $x^y$  は  $x$  の  $y$  乗を示す）。

### 【請求項 7】

前記部分群 S は、 $\mathbb{Z}/p^m\mathbb{Z}$  の乗法群であることを特徴とする、請求項 6 記載の加算プログラム。

### 【請求項 8】

前記正整数  $m$  は 2 であることを特徴とする、請求項 6 記載の加算プログラム。

### 【請求項 9】

前記部分群 S は、アノマラス橙円曲線の群であることを特徴とする、請求項 1 記載の加算プログラム。

### 【請求項 10】

前記群 G は、二つのアノマラス橙円曲線の群の直積であることを特徴とする、請求項 1 記載の加算プログラム。

### 【請求項 11】

さらに、前記逆変換モジュールは、予め一種類以上の数を幂乗もしくは幂倍して幂演算した結果を格納した格納手段を備えることを特徴とする、請求項 1 から請求項 10 のいずれか 1 項に記載の加算プログラム。

### 【請求項 12】

さらに、前記逆変換モジュールは、前記群 G に属する元を前記部分群 S に属する元に還元する還元手段を備えることを特徴とする、請求項 1 から請求項 10 のいずれか 1 項に記載の加算プログラム。

### 【請求項 13】

平文と鍵から暗号文を計算する暗号化プログラムであって、

整数同士の加算を行う一以上の加算モジュールを備え、

前記加算モジュールの一部または全部を請求項1から請求項12のいずれか1項に記載の加算プログラムを処理して実行することを特徴とする、暗号化プログラム。

#### 【請求項14】

前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項13記載の暗号化プログラム。

#### 【請求項15】

前記暗号文は共通鍵暗号を用いて計算することを特徴とする、請求項13または請求項14記載の暗号化プログラム。

#### 【請求項16】

暗号文と鍵から平文を計算する復号化プログラムであって、

整数同士の加算を行う一以上の加算モジュールを備え、

前記加算モジュールの一部または全部を請求項1から請求項12のいずれか1項に記載の加算プログラムを処理して実行することを特徴とする、復号化プログラム。

#### 【請求項17】

前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項16記載の復号化プログラム。

#### 【請求項18】

前記平文は共通鍵暗号を用いて計算することを特徴とする、請求項16または請求項17記載の復号化プログラム。

#### 【請求項19】

データに対し、デジタル署名を生成する署名生成プログラムであって、

整数同士の加算を行う一以上の加算モジュールを備え、

前記加算モジュールの一部または全部を請求項1から請求項12のいずれか1項に記載の加算プログラムを処理して実行することを特徴とする、署名生成プログラム。

#### 【請求項20】

前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項19記載の署名生成プログラム。

#### 【請求項21】

平文と鍵から暗号文を計算する暗号化装置であって、

整数同士の加算を行う一以上の加算演算部を備え、

前記加算演算部の一部または全部を請求項1から請求項12記載の加算プログラムを処理して実行することを特徴とする、暗号化装置。

#### 【請求項22】

前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項21記載の暗号化装置。

#### 【請求項23】

暗号文と鍵から平文を計算する復号化装置であって、

整数同士の加算を行う一以上の加算演算部を備え、

前記加算演算部の一部または全部を請求項1から請求項12のいずれか1項に記載の加算プログラムを処理して実行することを特徴とする、復号化装置。

#### 【請求項24】

前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項23記載の復号化装置。

#### 【請求項25】

群G用いて、2個以上の整数の加算を行う加算方法であって、

前記群Gは、真に含む部分群Sをもち、

前記群G上の幕演算を行うことにより、前記整数を前記Gに属する元に変換する変換ステップと、

前記群G上の基本演算を行う主要演算ステップと、

前記群 G または、前記群 G の部分群 S における前記変換ステップで行う変換の逆変換を行う逆変換ステップと、を含むことを特徴とする、加算方法。

【請求項 26】

請求項 1 から請求項 20 のいずれか 1 項に記載のプログラムを記録した記録媒体。

【請求項 27】

請求項 1 から請求項 20 のいずれか 1 項に記載のプログラムを実行する IC カード。

【書類名】明細書

【発明の名称】加算プログラム

【技術分野】

【0001】

本発明は、情報セキュリティ技術としてのソフトウェアの難読化方法に関する。

【背景技術】

【0002】

暗号ソフトを実装する場合、鍵や暗号アルゴリズムをそのまま実装すると、ソフトを解析された場合に、簡単に不正使用できる。そのため、ソフト解析を困難にする耐タンバーソフト技術が要望されている。耐タンバーソフト技術として、特許文献1において、演算及び演算領域を変換して、変換前の演算領域を推測困難にすることにより、ソフト解析困難とする方式が記載されている。

【0003】

その方式は、一次変換などにより変換を行う。例えば、鍵とデータの加算を変換する場合、鍵とデータを変換し、変換後のデータを変換後の領域における加算を行い、その結果に逆変換を実行することにより鍵とデータの加算結果を得る。このような難読化された加算方法を、共通鍵暗号を用いた暗号化プログラムや復号化プログラムに適用することにより、プログラムを解析して鍵を得る攻撃に対する安全性を向上させることができることになる。

【0004】

（従来例）

図6は、加算プログラム500の構成を示す図である。本プログラムは変換モジュール510、主要演算モジュール520、逆変換モジュール530、出力モジュール540から構成される。本プログラムは、入力a, bに対し、 $a + b$ を出力するものである。

変換モジュール510は、整数 $k_1, k_2$ を保持し、これらを用いて、入力a, bをそれぞれ $t_a = k_1 \times a + k_2, t_b = k_1 \times b + k_2$ に変換する。ただし、 $\times$ は乗算を示す。

次に主要演算モジュール520は、 $t_a, t_b$ に対して、 $t_{ab} = t_a + t_b$ を計算する。

逆変換モジュール530は、 $t_{ab}$ に対して、 $c = (t_{ab} - 2k_2) / k_1$ を計算する。

出力モジュール540は、 $c$ を出力する。

【0005】

上記のように処理すると、 $t_{ab} = t_a + t_b = k_1 \times a + k_2 + k_1 \times b + k_2 = k_1 \times (a + b) + 2k_2$ より、 $(t_{ab} - 2k_2) / k_1 = a + b$ が成り立つ。したがって、 $c = a + b$ となり、加算プログラム500によりaとbの加算結果が得られる。

ここで、変換モジュール510及び逆変換モジュール530を解析困難なように実現した場合、解析者が解析可能なものは、 $t_a, t_b, t_{ab}$ のみであり、これらの値からa, bを推測するのが難しいため、a, bを隠蔽することが可能になる。

【特許文献1】米国特許第6594761号明細書

【特許文献2】特許第3402441号明細書

【特許文献3】特許第2760799号明細書

【非特許文献1】岡本龍明、山本博資、「現代暗号」、産業図書（1997年）

【非特許文献2】Henri Cohen, "A Course in Computational Algebraic Number Theory", GTM 138, Springer-Verlag, 1993, pp. 19-20

【非特許文献3】I. Blake, G. Seroussi and N. Smart, "Elliptic Curves in Cryptography", CAMBRIDGE UNIVERSITY PRESS, 1999

【非特許文献4】N. Kunihiro and K. Koyama, "Two Discrete Log Algorithms for Super-An

【発明の開示】

【発明が解決しようとする課題】

【0006】

上記従来例の方式は、変換後の領域においても通常の加算と同様の演算を行うため、変換後の演算を解析することにより、変換前の演算が加算であることが判明しやすいという課題がある。プログラムの基となるアルゴリズムが解析者に既知の場合、解析者が加算をしている場所を分かると、その部分に集中して解析することにより、どのような変換を行っているか知られてしまう恐れがある。ゆえに、可能な限り、変換前の演算を知られないようにした方がよい。

【0007】

本発明は、変換前の演算を他の演算に変えるような変換を行うことにより、変換前の演算を解析しにくくするソースコード難読化方法を提供することを目的とする。

【課題を解決するための手段】

【0008】

上記目的を達成するために、請求項1における発明は、群G用いて、2個以上の整数の加算を行う加算プログラムであって、前記群Gは、真に含む部分群Sをもち、前記群G上の幕演算を行うことにより、前記整数を前記Gに属する元に変換する変換モジュールと、前記群G上の基本演算を行う主要演算モジュールと、前記群Gまたは、前記群Gの部分群Sにおける前記変換モジュールで行う変換の逆変換を行う逆変換モジュールと、を備えることを特徴とする。

【0009】

請求項2における発明は、前記群Gは、剩余整数環の乗法群であることを特徴とする。

請求項3における発明は、前記群Gは、複数個の相異なる素数  $p_1, p_2, \dots, p_k$  ( $k > 1$ ) の積  $n = p_1 \times p_2 \times \dots \times p_k$  に対し、 $\mathbb{Z}/n\mathbb{Z}$  の乗法群であることを特徴とする（ただし、 $\times$ は乗算である。また、 $\mathbb{Z}$ は整数環であり、 $\mathbb{Z}/n\mathbb{Z}$ は整数を  $m \bmod n$  した値で構成される剩余整数環である）。

【0010】

請求項4における発明は、前記逆変換モジュールは、前記素数  $p_1, p_2, \dots, p_k$  を用いた  $\mathbb{Z}/p_1\mathbb{Z}, \mathbb{Z}/p_2\mathbb{Z}, \dots, \mathbb{Z}/p_k\mathbb{Z}$  の乗法群における離散対数問題を解くことを特徴とする。

請求項5における発明は、前記逆変換モジュールは、前記素数  $p_1, p_2, \dots, p_k$  を用いた  $\mathbb{Z}/p_1\mathbb{Z}, \mathbb{Z}/p_2\mathbb{Z}, \dots, \mathbb{Z}/p_k\mathbb{Z}$  の乗法群における離散対数問題の解に対し、中国人の剩余定理を用いることを特徴とする。

【0011】

請求項6における発明は、前記群Gは、2つの素数  $p, q$  と正整数  $m$  を用いて表される  $n = p^m \times q$  に対し、 $\mathbb{Z}/n\mathbb{Z}$  の乗法群であることを特徴とする。（ただし、 $x^y$  は  $x$  の  $y$  乗を示す。）

請求項7における発明は、前記部分群Sは、 $\mathbb{Z}/p^m\mathbb{Z}$  の乗法群であることを特徴とする。

【0012】

請求項8における発明は、前記正整数  $m$  は2であることを特徴とする。

請求項9における発明は、前記部分群Sは、アノマラス橙円曲線の群であることを特徴とする。

請求項10における発明は、前記群Gは、二つのアノマラス橙円曲線の群の直積であることを特徴とする。

【0013】

請求項11における発明は、さらに、前記逆変換モジュールは、予め一種類以上の数を幕乗もしくは幕倍して幕演算した結果を格納した格納手段を備えることを特徴とする。

請求項 1 2 における発明は、さらに、前記逆変換モジュールは、前記群 G に属する元を前記部分群 S に属する元に還元する還元手段を備えることを特徴とする。

請求項 1 3 における発明は、平文と鍵から暗号文を計算する暗号化プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項 1 から請求項 1 2 のいずれか 1 項に記載の加算プログラムを処理して実行することを特徴とする。

#### 【0014】

請求項 1 4 における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。

請求項 1 5 における発明は、前記暗号文は共通鍵暗号を用いて計算することを特徴とする。

請求項 1 6 における発明は、暗号文と鍵から平文を計算する復号化プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項 1 から請求項 1 2 のいずれか 1 項に記載の加算プログラムを処理して実行することを特徴とする。

#### 【0015】

請求項 1 7 における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。

請求項 1 8 における発明は、前記平文は共通鍵暗号を用いて計算することを特徴とする。

請求項 1 9 における発明は、データに対し、デジタル署名を生成する署名生成プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項 1 から請求項 1 2 のいずれか 1 項に記載の加算プログラムを処理して実行することを特徴とする。

#### 【0016】

請求項 2 0 における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。

請求項 2 1 における発明は、平文と鍵から暗号文を計算する暗号化装置であって、整数同士の加算を行う一以上の加算演算部を備え、前記加算演算部の一部または全部を請求項 1 から請求項 1 2 のいずれか 1 項に記載の加算プログラムを処理して実行することを特徴とする。

#### 【0017】

請求項 2 2 における発明は、前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする。

請求項 2 3 における発明は、暗号文と鍵から平文を計算する復号化装置であって、整数同士の加算を行う一以上の加算演算部を備え、前記加算演算部の一部または全部を請求項 1 から請求項 1 2 のいずれか 1 項に記載の加算プログラムを処理して実行することを特徴とする。

#### 【0018】

請求項 2 4 における発明は、前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする。

請求項 2 5 における発明は、群 G 用いて、2 個以上の整数の加算を行う加算方法であって、前記群 G は、真に含む部分群 S をもち、前記群 G 上の幂演算を行うことにより、前記整数を前記 G に属する元に変換する変換ステップと、前記群 G 上の基本演算を行う主要演算ステップと、前記群 G または、前記群 G の部分群 S における前記変換ステップで行う変換の逆変換を行う逆変換ステップと、を含むことを特徴とする。

#### 【0019】

請求項 2 6 における発明は、請求項 1 から請求項 2 0 のいずれか 1 項に記載のプログラムを記録した記録媒体である。

請求項 2 7 における発明は、請求項 1 から請求項 2 0 のいずれか 1 項に記載のプログラ

ムを実行するＩＣカードである。

#### 【発明の効果】

##### 【0020】

これらの構成によると、演算に使用する値の隠蔽だけでなく、演算そのものを隠蔽することができ、その価値は大きい。

#### 【発明を実施するための最良の形態】

##### 【0021】

###### （実施の形態1）

本発明にかかる実施の形態1としての加算プログラム100について説明する。

図1は、実施の形態1における加算プログラム100の構成を示す図である。本プログラムは、変換モジュール110、主要演算モジュール120、逆変換モジュール130、出力モジュール140から構成される。本プログラムは、入力a, bに対し、 $a + b$ を出力するものである。

##### 【0022】

###### ＜各種パラメータ・記号の定義・入力の条件＞

ここで、実施の形態1としての加算プログラム100で使用する各種パラメータ・記号の定義を示す。

$p_i$  ( $i = 1, 2, \dots, k$ ) を素数とし、それらの積  $p_1 \times p_2 \times \dots \times p_k$  を  $n$  とする。ここで、 $\times$  は乗算を示す。 $p_i$  ( $i = 1, 2, \dots, k$ ) は逆変換モジュール130が保持し、 $n$  は変換モジュール110、主要演算モジュール120がそれぞれ保持する。

##### 【0023】

実施の形態1では、 $m \bmod n$  の整数から構成される剰余整数環  $\mathbb{Z}/n\mathbb{Z}$  の乗法群の演算を用いる。 $g$  は、その乗法群に属する予め与えられた数であり、 $p_i$  ( $i = 1, 2, \dots, k$ ) に対して原始元とする。原始元とは、 $m$  を  $1, 2, \dots$  と動かしたとき、初めて  $g^m \equiv 1 \pmod{p_i}$  となる  $m$  の値が  $p_i - 1$  であるような  $g$  である。

$L = \text{lcm}(p_1 - 1, p_2 - 1, \dots, p_k - 1)$  とする。ここで、 $\text{lcm}(p_1 - 1, p_2 - 1, \dots, p_k - 1)$  は  $p_1 - 1, p_2 - 1, \dots, p_k - 1$  の最小公倍数を示す。

##### 【0024】

入力  $a, b$  は、 $L/2$  より小さい非負整数とする。

###### ＜変換モジュール110の構成・処理＞

変換モジュール110は、 $n$  と  $g$  を格納する第1のパラメータ格納手段と、入力  $a, b$  に対し、 $ga = g^a \bmod n, gb = g^b \bmod n$  を計算する幂乗計算手段を備える。ここで、 $g^a$  は  $g$  の  $a$  乗を示す。

##### 【0025】

変換モジュール110は、入力  $a, b$  に対し、 $ga = g^a \bmod n, gb = g^b \bmod n$  を計算し、それらの計算結果  $ga, gb$  を主要演算モジュール120へ入力する。

###### ＜主要演算モジュール120の構成・処理＞

主要演算モジュール120は、 $n$  を格納する第2のパラメータ格納手段と、入力  $ga, gb$  に対し、 $gab = ga \times gb \bmod n$  を計算する乗算手段を備える。

##### 【0026】

主要演算モジュール120は、入力  $ga, gb$  に対し、第2のパラメータ格納手段に格納されている  $n$  を用いて、 $gab = ga \times gb \bmod n$  を計算し、その計算結果  $gab$  を逆変換モジュール130へ入力する。

###### ＜逆変換モジュール130の構成・処理＞

逆変換モジュール130は、 $p_i$  ( $i = 1, 2, \dots, k$ ) を格納する第3のパラメータ格納手段と、入力  $gab$  に対し、 $gab \bmod p_i$  ( $i = 1, 2, \dots, k$ ) の  $g \bmod p_i$  に対する離散対数  $c_i \bmod p_i - 1$  を計算する離散対数計算手段と、離散対数計算手段で計算した  $i = 1, 2, \dots, k$  に対する各  $c_i$  から、 $gab \bmod n$  の  $g \bmod n$  に対する離散対数  $c \bmod L$  を求める中国人剰余定理利用手段を備

える。

### 【0027】

逆変換モジュール130は、まず、入力 $g \ a \ b$ に対し、 $i = 1, 2, \dots, k$ に対して、 $g \ a \ b \ \text{mod} \ p_i$ の $g \ \text{mod} \ p_i$ に対する離散対数 $c_i \ \text{mod} \ p_i - 1$ を計算する。次に、 $i = 1, 2, \dots, k$ に対する各 $c_i$ を用いて、中国人の剰余定理 (Chinese Remainder Theorem) を用いて、 $c \ \text{mod} \ L$ を計算し、 $c$ を出力モジュール140へ入力する。

### 【0028】

離散対数計算手段における $c_i \ \text{mod} \ p_i - 1$ の計算方法については、様々なものがあるが、以下でその一例を示す。

$w$ を $1, 2, 3, \dots$ と順番に動かして、 $g^w \equiv g \ a \ b \ \text{mod} \ p_i$ となる $w$ を求める。その $w$ を $c_i$ とする。なお、各 $p_i$ に関し、 $g^1, g^2, \dots, g^{(p_i-2)}$ を計算した結果をテーブルとして保持し、そのテーブルの値から、 $g \ a \ b \ \text{mod} \ p_i$ と一致する $g^w$ を探す、としてもよい。

### 【0029】

中国人剰余定理利用手段における $c_i$  ( $i = 1, 2, \dots, k$ ) から、 $c \ \text{mod} \ p_i - 1 = c_i$ を満たす $c \ \text{mod} \ L$ を計算する方法については、非特許文献2が詳しい。

#### ＜出力モジュール140の構成・処理＞

出力モジュール140は、プログラムの外部へ数値を出力する出力手段を備える。

### 【0030】

出力モジュール140は、入力 $c$ をプログラムの外部へ出力する。

#### ＜実施の形態1の全体の動作＞

実施の形態1における加算プログラム100の全体の動作を以下に示す。

加算プログラム100の変換モジュール110は、入力 $a, b$ に対し $g_a, g_b$ を計算する。次に主要演算モジュール120は、 $g_a, g_b$ に対し、 $g_a b$ を計算する。さらに、逆変換モジュール130は、 $g_a b$ に対し、 $c$ を計算し、出力モジュール140は、 $c$ を出力する。

### 【0031】

#### ＜実施の形態1の動作検証＞

以下で、実施の形態1における加算プログラム100が、入力 $a, b$ に対し、 $a + b$ を出力していることを検証する。

変換モジュール110において、 $a, b$ に対し、 $g_a = g^a \ \text{mod} \ n, g_b = g^b \ \text{mod} \ n$ を計算し、主要演算ステップモジュール120で $g_a b = g_a \times g_b \ \text{mod} \ n$ を計算する。このとき、 $g_a b = g^{(a+b)} \ \text{mod} \ n$ を満たすことは明らかである。逆変換モジュール130では、 $g$ と $g_a b$ から $g_a b = g^{c_i} \ \text{mod} \ p_i$  ( $i = 1, 2, \dots, k$ ) を満たす $c_i$ を計算し、その結果を用いて、 $c = c_i \ \text{mod} \ p_i - 1$ を満たす $c \ \text{mod} \ L$ を計算する。このとき、 $c$ は $g_a b = g^c \ \text{mod} \ n$ を満たす。なぜなら、 $a + b = c \ \text{mod} \ L$ より、 $g^{(a+b-c)} = 1 \ \text{mod} \ n$ となるためである。したがって、 $g^{(a+b)} \ \text{mod} \ n = g^c \ \text{mod} \ n$ を満たすため、 $a + b = c \ \text{mod} \ ((p_1 - 1) \times (p_2 - 1) \times \dots \times (p_k - 1))$ を満たす。 $a < L/2, b < L/2$ より $a + b < L$ であるので、加算プログラム100は $a$ と $b$ の加算結果 $a + b$ を出力していることになる。

### 【0032】

#### ＜実施の形態1の効果＞

実施の形態1における加算プログラム100は、加算を行う値を変換しており、変換モジュール110及び逆変換モジュール130が解析困難な場合に、解析者は $g_a, g_b, g_a b$ の値と、 $g_a, g_b$ から $g_a b$ を計算する処理を解析可能である。そのとき、変換後の値 $g_a, g_b$ から変換前の値 $a, b$ を推測することは困難である。さらに、加算プログラム100は、主要演算モジュール120において、乗算を行っており、この乗算とい

う演算から加算プログラム100が加算を実現していることを推測することは困難である。したがって、加算を行う入力の値の隠蔽だけではなく、加算という演算自体も隠蔽できることになり、本実施の形態1は有効である。

### 【0033】

#### （実施の形態2）

本発明にかかる実施の形態2としての加算プログラム200について説明する。

図2は、実施の形態2における加算プログラム200の構成を示す図である。本プログラムは、実施の形態1と同様の出力モジュール140と、実施の形態1と異なる変換モジュール210、主要演算モジュール220、逆変換モジュール230から構成される。本プログラムは、入力a, bに対し、 $a + b$ を出力するものである。

### 【0034】

#### （各種パラメータ・記号の定義・入力の条件）

ここで、実施の形態2としての加算プログラム200で使用する各種パラメータ・記号の定義を示す。

p, qを素数とし、 $n = p^2 \times q$ とする。p, qは逆変換モジュール230が保持し、nは変換モジュール210、主要演算モジュール220がそれぞれ保持する。

### 【0035】

実施の形態2では、 $m \bmod n$ の整数から構成される剰余整数環 $\mathbb{Z}/n\mathbb{Z}$ の乗法群の演算を用いる。gは、その乗法群に属する予め与えられた数であり $g^{(p-1)} \bmod p^2$ の位数がpである数とする。 $g^p = g^{(p-1)} \bmod p^2$ と定義する。

入力a, bは、 $p/2$ より小さい非負整数とする。

### 【0036】

#### （変換モジュール210の構成・処理）

変換モジュール210は、nとgを格納する第1のパラメータ格納手段と、n以下の乱数R1, R2を発生する乱数発生手段と、入力a, bに対し、乱数発生手段が発生した乱数R1, R2を用いて、 $g^a = g^{(a + n \times R1)} \bmod n$ ,  $g^b = g^{(b + n \times R2)} \bmod n$ を計算する幕乗計算手段を備える。

### 【0037】

変換モジュール210は、乱数R1, R2を発生して、入力a, bに対し、 $g^a = g^{(a + n \times R1)} \bmod n$ ,  $g^b = g^{(b + n \times R2)} \bmod n$ を計算し、それらの計算結果 $g^a$ ,  $g^b$ を主要演算モジュール220へ入力する。

#### （主要演算モジュール220の構成・処理）

主要演算モジュール220は、nを格納する第2のパラメータ格納手段と、入力 $g^a$ ,  $g^b$ に対し、 $g^a \times g^b = g^{a+b} \bmod n$ を計算する乗算手段を備える。

### 【0038】

主要演算モジュール220は、入力 $g^a$ ,  $g^b$ に対し、第2のパラメータ格納手段に格納されているnを用いて、 $g^a \times g^b = g^{a+b} \bmod n$ を計算し、その計算結果 $g^a \times g^b$ を逆変換モジュール230へ入力する。

#### （逆変換モジュール230の構成・処理）

逆変換モジュール230は、pを格納する第3のパラメータ格納手段と、入力 $g^a \times g^b$ に対し、第3のパラメータ格納手段に格納されているpを用いて、 $c_p = g^{a+b} \bmod p^2$ を計算する還元手段と、 $m \bmod p^2$ における $g^p$ に対する $c_p$ の離散対数を計算する離散対数計算手段を備える。

### 【0039】

逆変換モジュール230は、まず、入力 $g^a \times g^b$ に対し、 $c_p = g^{a+b} \bmod p^2$ を計算する。次に、 $m \bmod p^2$ における $g^p$ に対する $c_p$ の離散対数cを計算し、cを出力モジュール240へ入力する。

離散対数計算手段におけるcの計算方法は、特許文献2が詳しい。

具体的には、以下のように行う。

#### 【0040】

$c p$  に対し、 $c = (c p - 1) / (g - 1) \bmod p$  として  $c$  を求める。

＜実施の形態2の全体の動作＞

実施の形態2における加算プログラム200の全体の動作を以下に示す。

加算プログラム200の変換モジュール210は、入力  $a, b$  に対し  $g a, g b$  を計算する。次に主要演算モジュール220は、 $g a, g b$  に対し、 $g a b$  を計算する。さらに、逆変換モジュール230は、 $g a b$  に対し、 $c$  を計算し、出力モジュール140は、 $c$  を出力する。

#### 【0041】

＜実施の形態2の動作検証＞

以下で、実施の形態2における加算プログラム200が、入力  $a, b$  に対し、 $a + b$  を出力していることを検証する。

変換モジュール210において、 $a, b$  に対し、 $g a = g^{\wedge} (a + n \times R1) \bmod n, g b = g^{\wedge} (b + n \times R2) \bmod n$  を計算し、主要演算ステップモジュール220で  $g a b = g a \times g b \bmod n$  を計算する。このとき、 $g a b = g^{\wedge} (a + b + n \times (R1 + R2)) \bmod n$  を満たすことは明らかである。逆変換モジュール230では、まず、 $c p = g a b^{\wedge} (p - 1) = g p^{\wedge} (a + b + n \times (R1 + R2)) \bmod p^{\wedge} 2$  となり、 $g p^{\wedge} p = 1 \bmod p^{\wedge} 2$  より、 $g p^{\wedge} n = 1 \bmod p^{\wedge} 2$  であるため、 $c p = g p^{\wedge} (a + b) \bmod p^{\wedge} 2$  となる。

#### 【0042】

逆変換モジュール230では、さらに、 $\bmod p^{\wedge} 2$  における  $g p$  に対する  $c p$  の離散対数  $c$  を求める。すなわち、 $c p = g p^{\wedge} c \bmod p^{\wedge} 2$  が成り立つ。したがって、 $c = a + b \bmod p$  であり、 $a < p/2, b < p/2$  より、 $a + b < p$  であるため、加算プログラム200は  $a$  と  $b$  の加算結果  $a + b$  を出力していることになる。

＜実施の形態2の効果＞

実施の形態2における加算プログラム200は、実施の形態1と同様に、加算を行う値を変換しており、変換モジュール210及び逆変換モジュール230が解析困難な場合に、変換後の値から変換前の値を推測することは困難である。さらに、加算プログラム200は、主要演算モジュール220において、乗算を行っており、この乗算という演算から加算プログラム200が加算を実現していることを推測することは困難である。したがって、加算を行う入力の値の隠蔽だけではなく、加算という演算自体も隠蔽できることになり、本実施の形態2は有効である。

#### 【0043】

実施の形態2における加算プログラム200では、剩余整数環  $\mathbb{Z}/n\mathbb{Z}$  の乗法群における幂乗演算を変換モジュールで行い、その乗法群の部分群である剩余整数環  $\mathbb{Z}/p^{\wedge} 2$  の乗法群における離散対数問題を逆変換モジュールで解いている。ここで、もし、解析者が  $p, q$  は分からないが、変換モジュールで幂乗演算を行っていることを解析できた場合を考える。このケースは、すなわち、逆変換モジュールのみが解析者により解析困難である場合である。この場合においても、 $n$  の大きさが素因数分解が困難なぐらい、例えは、1024ビットぐらいであれば、 $n$  の素因数分解結果である  $p, q$  を得ることが困難になる。また、 $p, q$  が得られなければ、剩余整数環  $\mathbb{Z}/n\mathbb{Z}$  の乗法群における離散対数問題を解くことが困難になる。一般に乗法群の大きさ（元の数）が1024ビットの数のように大きい場合は、それ上の離散対数問題も困難になる。実施の形態2では、 $p$  が既知の場合は逆変換モジュールにおける逆変換の方法によって、 $\mathbb{Z}/p^{\wedge} 2$  の乗法群における離散対数問題を容易に解けるようになる。実施の形態2における変換はこのように、 $p$  が既知であれば逆変換が容易であるが、既知でなければ困難であることを利用している点が、実施の形態1と異なる。

#### 【0044】

（実施の形態3）

本発明にかかる実施の形態3としての加算プログラム300について説明する。本加算

プログラム300は、橙円曲線のスカラ倍演算を利用している。橙円曲線については、非特許文献3が詳しい。

図3は、実施の形態3における加算プログラム300の構成を示す図である。本プログラムは、実施の形態1と同様の出力モジュール140と、実施の形態1と異なる変換モジュール310、主要演算モジュール320、逆変換モジュール330から構成される。本プログラムは、入力a, bに対し、 $a + b$ を出力するものである。

#### 【0045】

＜各種パラメータ・記号の定義、入力の条件＞

ここで、実施の形態3としての加算プログラム300で使用する各種パラメータ・記号の定義を示す。

$p, q$ を素数とし、 $n = p \times q$ とする。 $p, q$ は逆変換モジュール330が保持し、 $n$ は変換モジュール310、主要演算モジュール320がそれぞれ保持する。

#### 【0046】

橙円曲線Eの方程式を $y^2 = x^3 + Ax + B$ とする。 $A, B$ はパラメータである。 $G = (x_g, y_g) \bmod n$ を橙円曲線E上の点とする。すなわち、 $y_g^2 = x_g^3 + Ax_g + B \bmod n$ を満たす。 $A, B, G$ は、変換モジュール310、主要演算モジュール320、逆変換モジュール330が保持する。

橙円曲線Eの方程式をもつ体 $GF(p)$ 上の橙円曲線の点から構成される群を $E(GF(p))$ と書く。同様に、橙円曲線Eの方程式をもつ体 $GF(q)$ 上の橙円曲線の点から構成される群を $E(GF(q))$ と書く。 $Z/nZ$ 上の橙円曲線の群を $E(GF(p))$ と $E(GF(q))$ の直積 $E(GF(p)) \times E(GF(q))$ で表す。 $Z/nZ$ は体ではなく、環であるため、数学的には橙円曲線とはよべないが、ここでは便宜上、その直積 $Z/nZ$ 上の橙円曲線の群とよぶ。

#### 【0047】

$E(GF(p))$ 上の点 $G_p = (x_g p, y_g p) \bmod p$ と、 $E(GF(q))$ 上の点 $G_q = (x_g q, y_g q) \bmod q$ に対応する $Z/nZ$ 上の橙円曲線 $E(GF(p)) \times E(GF(q))$ の点 $G = (x_g, y_g) \bmod n$ は、以下のように定義する。 $x_g$ を $x_g \bmod p = x_g p$ ,  $x_g \bmod q = x_g q$ を満たす数、 $y_g$ を $y_g \bmod p = y_g p$ ,  $y_g \bmod q = y_g q$ を満たす数とする。この定義より、 $E(GF(p)) \times E(GF(q))$ 上の点 $G = (x_g, y_g) \bmod n$ に対応する $E(GF(p))$ 上の点 $G_p$ を $G_p = (x_g p, y_g p) \bmod p$ とし、 $E(GF(q))$ 上の点 $G_q$ を $G_q = (x_g q, y_g q)$ とすることで、 $E(GF(p)), E(GF(q))$ を $E(GF(p)) \times E(GF(q))$ の部分群とみなす。

#### 【0048】

実施の形態3においては、上記橙円曲線Eは $\bmod p$ での橙円曲線の位数、すなわち、点の個数が $p$ であるとする。このような体 $GF(p)$ 上の橙円曲線をアノマラス(Anomalous)橙円曲線とよぶ。さらに、 $\bmod q$ での橙円曲線の位数が $q$ である、すなわち、 $GF(q)$ 上でもアノマラス橙円曲線であるとする。このとき、 $Z/nZ$ 上の橙円曲線は、スーパーアノマラス(Super-Anomalous)橙円曲線とよぶ。スーパーアノマラス橙円曲線については非特許文献4が詳しい。このとき、 $Z/nZ$ 上の橙円曲線の群は $E(GF(p)) \times E(GF(q))$ であるので、橙円曲線の位数は $n (= p \times q)$ となる。

#### 【0049】

入力 $a, b$ は、 $p/2$ より小さい非負整数とする。

＜変換モジュール310の構成・処理＞

変換モジュール310は、 $n, A, B, G$ を格納する第1のパラメータ格納手段と、入力 $a, b$ に対し、第1のパラメータ格納手段に格納されている $n, A, B, G$ を用いて、 $G_a = a * G \bmod n, G_b = b * G \bmod n$ を計算するスカラ倍演算手段を備える。ここで、 $a * G$ は、 $G$ を $a$ 回橙円曲線の加算により足し合させた点である。また、 $a * G \bmod n$ とは、 $a * G$ の各座標を $\bmod n$ したものである。

## 【0050】

変換モジュール310は、入力a, bに対し、 $G_a = a * G \bmod n$ ,  $G_b = b * G \bmod n$ を計算し、それらの計算結果 $G_a$ ,  $G_b$ を主要演算モジュール320へ入力する。

### ＜主要演算モジュール320の構成・処理＞

主要演算モジュール320は、n, A, Bを格納する第2のパラメータ格納手段と、入力 $G_a$ ,  $G_b$ に対し、 $G_{ab} = G_a + G_b \bmod n$ を計算する橙円曲線加算手段を備える。

## 【0051】

主要演算モジュール320は、入力 $G_a$ ,  $G_b$ に対し、第2のパラメータ格納手段に格納されているn, A, Bを用いて、橙円曲線加算を実行し、 $G_{ab} = G_a + G_b \bmod n$ を計算し、その計算結果 $G_{ab}$ を逆変換モジュール330へ入力する。

### ＜逆変換モジュール330の構成・処理＞

逆変換モジュール330は、p, A, B, G  $\bmod p$ を格納する第3のパラメータ格納手段と、入力 $G_{ab}$ に対し、第3のパラメータ格納手段に格納されているpを用いて、 $G_p = G_{ab} \bmod p$ を計算する還元手段と、 $G \bmod p$ に対する $G_p$ の離散対数を計算する橙円離散対数計算手段を備える。

## 【0052】

逆変換モジュール330は、まず、入力 $G_{ab}$ に対し、 $G_p = G_{ab} \bmod p$ を計算する。次に、 $G \bmod p$ に対する $G_p$ の離散対数 $c \bmod p$ を計算し、cを出力モジュール140へ入力する。

橙円離散対数計算手段におけるcは、アノマラス橙円曲線上の離散対数問題の解である。アノマラス橙円曲線上の離散対数問題を解く方法は、非特許文献3の88~91ページが詳しい。計算方法はこの文献に記載されているため、ここでは説明を省略する。

## 【0053】

### ＜実施の形態3の全体の動作＞

実施の形態3における加算プログラム300の全体の動作を以下に示す。

加算プログラム300の変換モジュール310は、入力a, bに対し $G_a$ ,  $G_b$ を計算する。次に主要演算モジュール320は、 $G_a$ ,  $G_b$ に対し、 $G_{ab}$ を計算する。さらに、逆変換モジュール330は、 $G_{ab}$ に対し、cを計算し、出力モジュール140は、cを出力する。

## 【0054】

### ＜実施の形態3の動作検証＞

以下で、実施の形態3における加算プログラム300が、入力a, bに対し、 $a + b$ を出力していることを検証する。

変換モジュール310において、a, bに対し、 $G_a = a * G \bmod n$ ,  $G_b = b * G \bmod n$ を計算し、主要演算モジュール320で、 $G_{ab} = G_a + G_b \bmod n$ を計算する。このとき、 $G_{ab} = (a + b) * G \bmod n$ を満たすことは明らかである。逆変換モジュール330では、まず、 $G_p = G_{ab} \bmod p$ を計算し、 $G \bmod p$ に対する $G_p$ の離散対数cを求める。すなわち、 $G_p = c * G \bmod p$ が成り立つ。したがって、 $c = a + b \bmod p$ であり、 $a < p/2$ ,  $b < p/2$ より、 $a + b < p$ であるため、加算プログラム300は、aとbの加算結果 $a + b$ を出力していることになる。

## 【0055】

### ＜実施の形態3の効果＞

実施の形態3における加算プログラム300は、実施の形態1, 2と同様に、加算を行う値を変換しており、変換モジュール310及び逆変換モジュール330が解析困難な場合に、変換後の値から変換前の値を推測することは困難である。さらに、加算プログラム300は、主要演算モジュール320において、橙円曲線加算を行っており、この橙円曲線加算という演算から加算プログラム300が整数の加算を実現していることを推測する

ことは困難である。したがって、整数の加算を行う入力の値の隠蔽だけではなく、整数の加算という演算自体も隠蔽できることになり、本実施の形態3は有効である。

#### 【0056】

実施の形態3における加算プログラム300では、 $Z/nZ$ 上の楕円曲線の群E( $GF(p)$ ) $\times$ E( $GF(q)$ )のスカラ倍演算を変換モジュールで行い、その部分群であるE( $GF(p)$ )における離散対数問題を逆変換モジュールで解いている。ここで、もし、解析者が $p$ ,  $q$ は分からぬが、変換モジュールで冪乗演算を行っていることを解析できた場合を考える。このケースは、すなわち、逆変換モジュールのみが解析者により解析困難である場合である。この場合においても、 $n$ の大きさが素因数分解が困難なぐらい、例えば、1024ビットぐらいであれば、 $n$ の素因数分解結果である $p$ ,  $q$ を得ることが困難になる。また、 $p$ ,  $q$ が得られなければ、 $Z/nZ$ 上の楕円曲線の群E( $GF(p)$ ) $\times$ E( $GF(q)$ )における離散対数問題を解くことが困難になる。一般に群の大きさ(元の数)が1024ビットの数のように大きい場合は、それ上の離散対数問題も困難になる。実施の形態3では、 $p$ が既知の場合は逆変換モジュールにおける逆変換の方法によって、楕円曲線の群E( $GF(p)$ )における離散対数問題を容易に解けるようになる。実施の形態3における変換はこのように、 $p$ が既知であれば逆変換が容易であるが、既知でなければ困難であることを利用している点が、実施の形態1と異なる。

#### 【0057】

##### (実施の形態4)

本発明に係る実施の形態4としての暗号化プログラム400について説明する。本暗号化プログラムは、特許文献3に記載されている暗号化方法に対し、鍵と平文または、暗号化の途中結果との加算の部分で、実施の形態1の加算プログラムを適用したものである。

図4は、実施の形態4における暗号化プログラム400の構成を示す図である。本プログラムは、拡張鍵生成モジュール410と、鍵加算モジュール420と、ローテーションモジュールA430と、ローテーションモジュールB440と、ローテーションモジュールC450と、ローテーションモジュールD460と、排他的論理和モジュール470と、ビット分割モジュール480と、出力モジュール490を備える。

#### 【0058】

暗号化プログラム400は、64ビットの鍵Kと、64ビットの平文Mに対し、64ビットの暗号文Cを出力する。

##### <拡張鍵生成モジュール410の処理>

拡張鍵生成モジュール410は、鍵Kを用いて、32ビットの8個の拡張鍵 $K_1, K_2, K_3, \dots, K_8$ を生成し、出力する。生成する方法については、特許文献3に記載されているため、説明を省略する。

#### 【0059】

##### <鍵加算モジュール420の処理>

鍵加算モジュール420は、実施の形態1の加算プログラムにより入力された2つの数の加算を実行し、加算結果を出力する。

##### <ローテーションモジュールA430の処理>

ローテーションモジュールA430は、入力Xに対し、 $Rot_2(X) + X + 1$ を計算し、出力する。ここで、 $Rot_2$ は、左へ2ビット循環シフトすることを示す。

#### 【0060】

##### <ローテーションモジュールB440の処理>

ローテーションモジュールB440は、入力Xに対し、 $Rot_4(X) \text{ XOR } X$ を計算し、出力する。ここで、 $Rot_4$ は、左へ4ビット循環シフトすることを示し、XOR排他的論理和を示す。

##### <ローテーションモジュールC450の処理>

ローテーションモジュールC450は、入力Xに対し、 $Rot_8(X) \text{ XOR } X$ を計算し、出力する。ここで、 $Rot_8$ は左へ8ビット循環シフトすることを示す。

#### 【0061】

### ＜ローテーションモジュール D 4 6 0 の処理＞

ローテーションモジュール D 4 6 0 は、入力 X, Y に対し、 $\text{Rot16}(X) + (X \text{ AND } Y)$  を計算し、出力する。ここで、 $\text{Rot16}$  は、左へ 16 ビット循環シフトすることを示し、 $\text{AND}$  は論理積を示す。

### ＜排他的論理和モジュール 4 7 0 の処理＞

排他的論理和モジュール 4 7 0 は、入力 X, Y に対し、 $X \text{ XOR } Y$  を計算し、出力する。

### 【0 0 6 2】

#### ＜ビット分割モジュール 4 8 0 の処理＞

ビット分割モジュール 4 8 0 は、入力である 64 ビットの数 X に対し、X の上位ビット 32 ビット X 1、下位ビット 32 ビット X 2 を出力する。

#### ＜出力モジュール 4 9 0 の処理＞

出力モジュール 4 9 0 は、入力である 32 ビットの数 X, Y に対し、X を上位、Y を下位とする 64 ビット整数を生成し、プログラムの外部へ出力する。

### 【0 0 6 3】

#### ＜暗号化プログラム 4 0 0 の処理＞

暗号化プログラム 4 0 0 の処理を以下に示す。図 5 は、暗号化プログラム 4 0 0 の処理を示すフローチャートである。

ステップ S 1 0 1：拡張鍵生成モジュール 4 1 0 は、入力の鍵 K から K 1, K 2, …, K 8 を生成する。

### 【0 0 6 4】

ステップ S 1 0 2：分割モジュール 4 9 0 は、入力の平文 M を上位 M 1 と下位 M 2 に分割する。

ステップ S 1 0 3：排他的論理和モジュール 4 7 0 は、M 1 と M 2 に対し、 $\text{TMP1} = M_1 \text{ XOR } M_2$  を計算する。

ステップ S 1 0 4：鍵加算モジュール 4 2 0 は、 $\text{TMP1}, K_1$  に対し、 $\text{TMP2} = \text{TMP1} + K_1$  を計算する。

### 【0 0 6 5】

ステップ S 1 0 5：ローテーションモジュール A 4 3 0 は、 $\text{TMP2}$  に対し、 $\text{TMP3} = \text{Rot2}(\text{TMP2}) + \text{TMP2} + 1$  を計算する。

ステップ S 1 0 6：ローテーションモジュール C 4 5 0 は、 $\text{TMP3}$  に対し、 $\text{TMP4} = \text{Rot4}(\text{TMP3}) \text{ XOR } \text{TMP3}$  を計算する。

ステップ S 1 0 7：排他的論理和モジュール 4 7 0 は、 $\text{TMP4}, M_1$  に対し、 $\text{TMP5} = \text{TMP4} \text{ XOR } M_1$  を計算する。

### 【0 0 6 6】

ステップ S 1 0 8：鍵加算モジュール 4 2 0 は、 $\text{TMP5}, K_2$  に対し、 $\text{TMP6} = \text{TMP5} + K_2$  を計算する。

ステップ S 1 0 9：ローテーションモジュール A 4 3 0 は、 $\text{TMP6}$  に対し、 $\text{TMP7} = \text{Rot2}(\text{TMP6}) + \text{TMP6} + 1$  を計算する。

ステップ S 1 1 0：ローテーションモジュール B 4 4 0 は、 $\text{TMP7}$  に対し、 $\text{TMP8} = \text{Rot8}(\text{TMP7}) + \text{TMP7} + 1$  を計算する。

### 【0 0 6 7】

ステップ S 1 1 1：鍵加算モジュール 4 2 0 は、 $\text{TMP8}$  と  $K_3$  に対し、 $\text{TMP9} = \text{TMP8} + K_3$  を計算する。

ステップ S 1 1 2：ローテーションモジュール A 4 3 0 は、 $\text{TMP9}$  に対し、 $\text{TMP10} = \text{Rot2}(\text{TMP9}) + \text{TMP9} + 1$  を計算する。

ステップ S 1 1 3：ローテーションモジュール D 4 6 0 は、 $\text{TMP10}$  と  $\text{TMP7}$  に対し、 $\text{TMP11} = \text{Rot16}(\text{TMP10}) + (\text{TMP10} \text{ AND } \text{TMP7})$  を計算する。

### 【0 0 6 8】

ステップS114：排他的論理和モジュール470は、TMP11とTMP1に対して、  
、 $TMPl2 = TMPl1 \text{ XOR } TMPl$ を計算する。

ステップS115：鍵加算モジュール420は、 $TMPl2$ と $K4$ に対し、 $TMPl3 = TMPl2 + K4$ を計算する。

ステップS116：ローテーションモジュールA430は、 $TMPl3$ に対し、 $TMPl4 = Rot2(TMPl3) + TMPl3 + 1$ を計算する。

#### 【0069】

ステップS117：排他的論理和モジュール470は、 $TMPl4$ と $TMPl4$ に対し、  
 $TMPl5 = TMPl4 \text{ XOR } TMPl4$ を計算する。

ステップS118：排他的論理和モジュール470は、 $TMPl5$ と $TMPl2$ に対し、  
 $TMPl6 = TMPl5 \text{ XOR } TMPl2$ を計算する。

ステップS119：出力モジュール490は、 $TMPl5$ を上位、 $TMPl6$ を下位とする64ビットの整数を生成し、暗号文Cとして出力する。

#### 【0070】

##### ＜実施の形態4の効果＞

実施の形態4における暗号化プログラムでは、鍵と平文または、暗号化の途中結果との加算において、実施の形態1の加算プログラムを利用する。そのため、加算を行う値、すなわち、鍵の値を推測することが困難になる。また、解析者が暗号アルゴリズムを知っている場合においても、鍵加算部分が、鍵との「加算」を行っていると推測しにくい。そのため、解析者が、暗号アルゴリズムの特徴である鍵加算部分をプログラム内から探し出す攻撃をした場合でも、鍵加算部分を探し出すことが困難なため、攻撃成功も困難となる。このように、解析者の攻撃が困難になり、本実施の形態4は有効である。

#### 【0071】

##### （変形例）

上記に説明した実施の形態は、本発明の実施の一例であり、本発明はこの実施の形態に何ら限定されるものではなく、その旨を逸脱しない範囲において種々なる態様で実施し得るものである。例えば、以下のような場合も本発明に含まれる。

（1）実施の形態1、2、3において、2個の非負整数 $a$ 、 $b$ の加算を行っていたが、これを3個またはそれ以上の個数の非負整数の加算を行ってもよい。その場合は、変換モジュールはそれぞれの非負整数を変換し、主要演算モジュールでは、それぞれの変換結果を実施の形態1、2では乗算、実施の形態3では橜円曲線加算を行う。

#### 【0072】

（2）暗号化プログラムにおいて、鍵加算部分に実施の形態1の加算プログラムを使用したが、これが実施の形態2または3の加算プログラムであってもよい。

（3）暗号化プログラムにおいて、鍵加算部分のみに加算プログラムを使用したが、他の加算部分についても使用してもよい。

（4）実施の形態4では、加算プログラムを暗号化プログラムに適用しているが、復号化プログラムや、デジタル署名を生成する署名生成プログラムに適用するとしてもよい。

#### 【0073】

（5）実施の形態1、2、3においては、剩余整数環の乗法群、橜円曲線上の群を利用したが、これを、その他の群を利用するとしてもよい。また、実施の形態1、2では幕乗演算、実施の形態3においては、橜円曲線のスカラ倍演算を行って、整数を変換したが、これをその他の群の幕演算としてもよい。幕演算とは、群の基本演算、すなわち、剩余整数環では乗算、橜円曲線上の群では橜円曲線加算を、整数回行った結果を求める演算である。したがって、剩余整数環の乗法群の幕演算は幕乗演算、橜円曲線上の群の幕演算は橜円曲線のスカラ倍演算である。実施の形態2では、剩余整数環 $\mathbb{Z}/n\mathbb{Z}$ の乗法群の「部分群」である、剩余整数環 $\mathbb{Z}/p^2\mathbb{Z}$ の乗法群において離散対数問題を解いている。その他の群を使用する場合は、実施の形態2と同様に逆変換モジュールでその他の群の「部分群」において離散対数問題を解いてもよい。

#### 【0074】

(6) 実施の形態1において、 $g$  は  $m \bmod p_i$  ( $i = 1, 2, \dots, k$ )において原始元としたが、原始元でなくてもよい。その場合は、 $g^m \bmod p_i$  ( $m > 0$ )となる  $m_i$  に対し、 $L = m_1 \times m_2 \times \dots \times m_k$  とする。

(7) これらの実施の形態及び変形例の組合せであってもよい。

### 【産業上の利用可能性】

#### 【0075】

これらの構成によると、演算に使用する値の隠蔽だけでなく、演算そのものを隠蔽することができる。したがって、本技術を用いた難読化ソフトウェアをICカード等の機器に組み込むことは有用である。

### 【図面の簡単な説明】

#### 【0076】

【図1】本発明に係る1個の実施の形態としての加算プログラム100の構成を示すブロック図

【図2】本発明に係る1個の実施の形態としての加算プログラム200の構成を示すブロック図

【図3】本発明に係る1個の実施の形態としての加算プログラム300の構成を示すブロック図

【図4】本発明に係る1個の実施の形態としての暗号化プログラム400の構成を示すブロック図

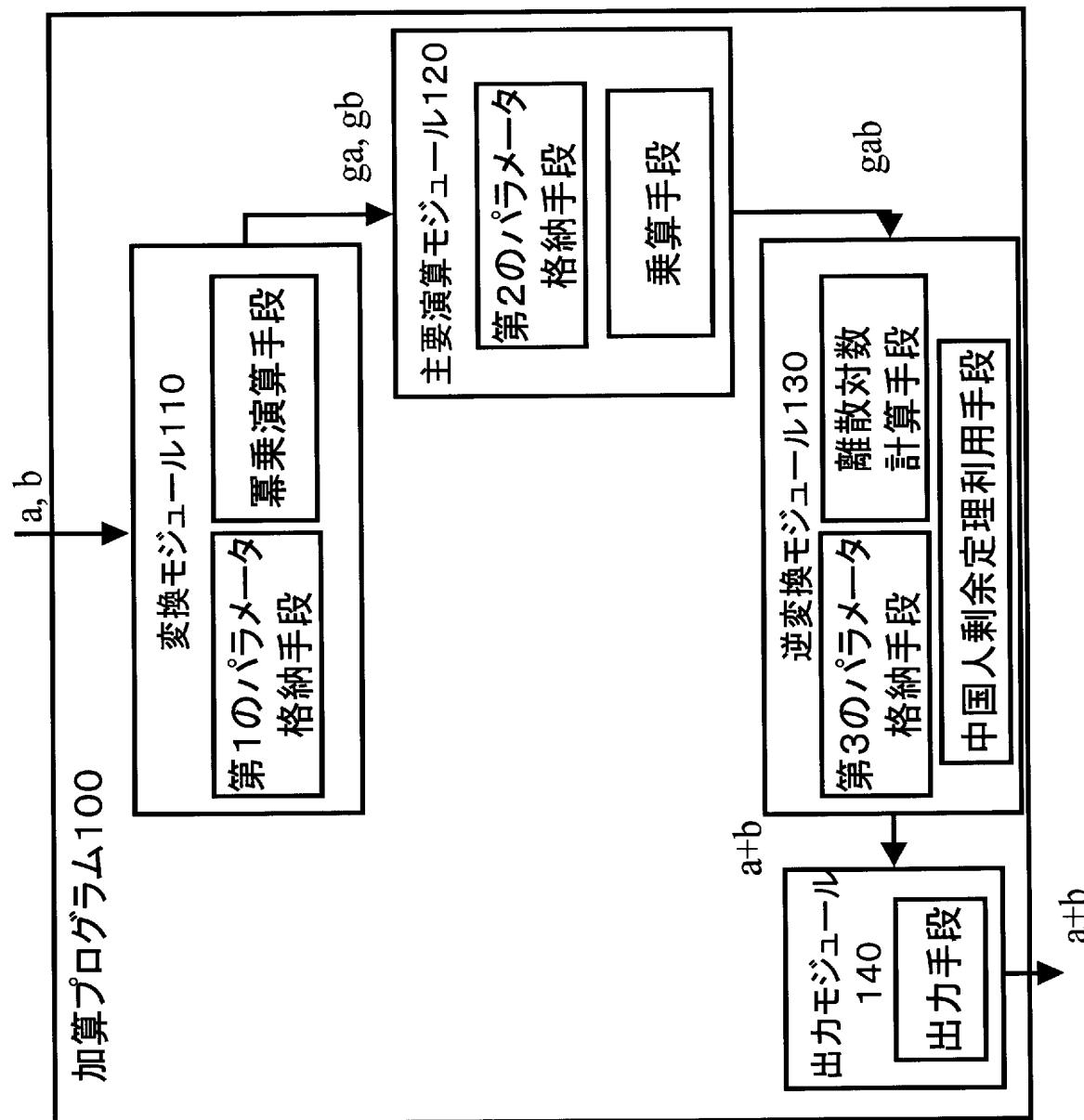
【図5】暗号化プログラム400の処理を示すフローチャート

【図6】従来例としての加算プログラム500の構成を示すブロック図

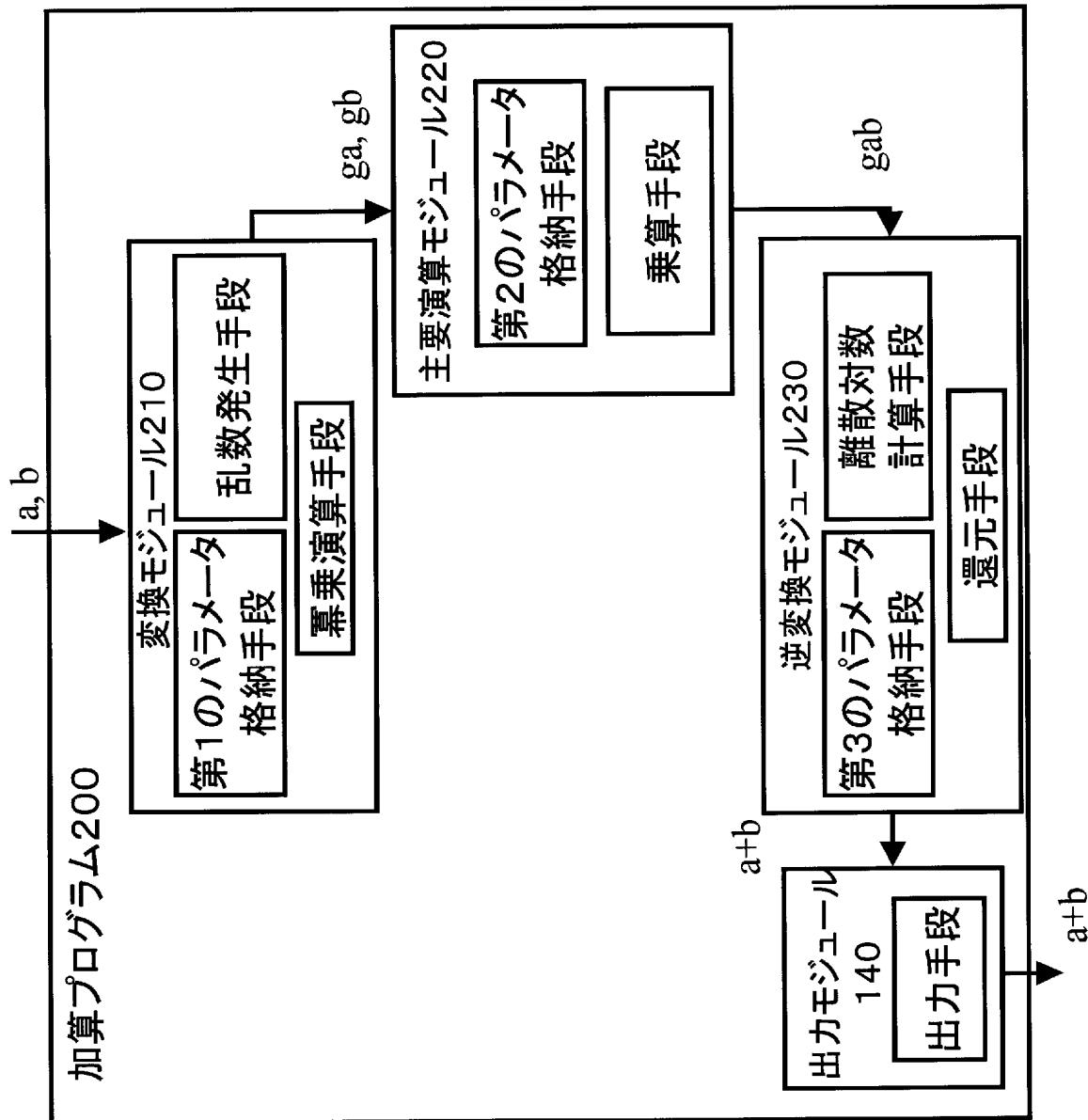
### 【符号の説明】

#### 【0077】

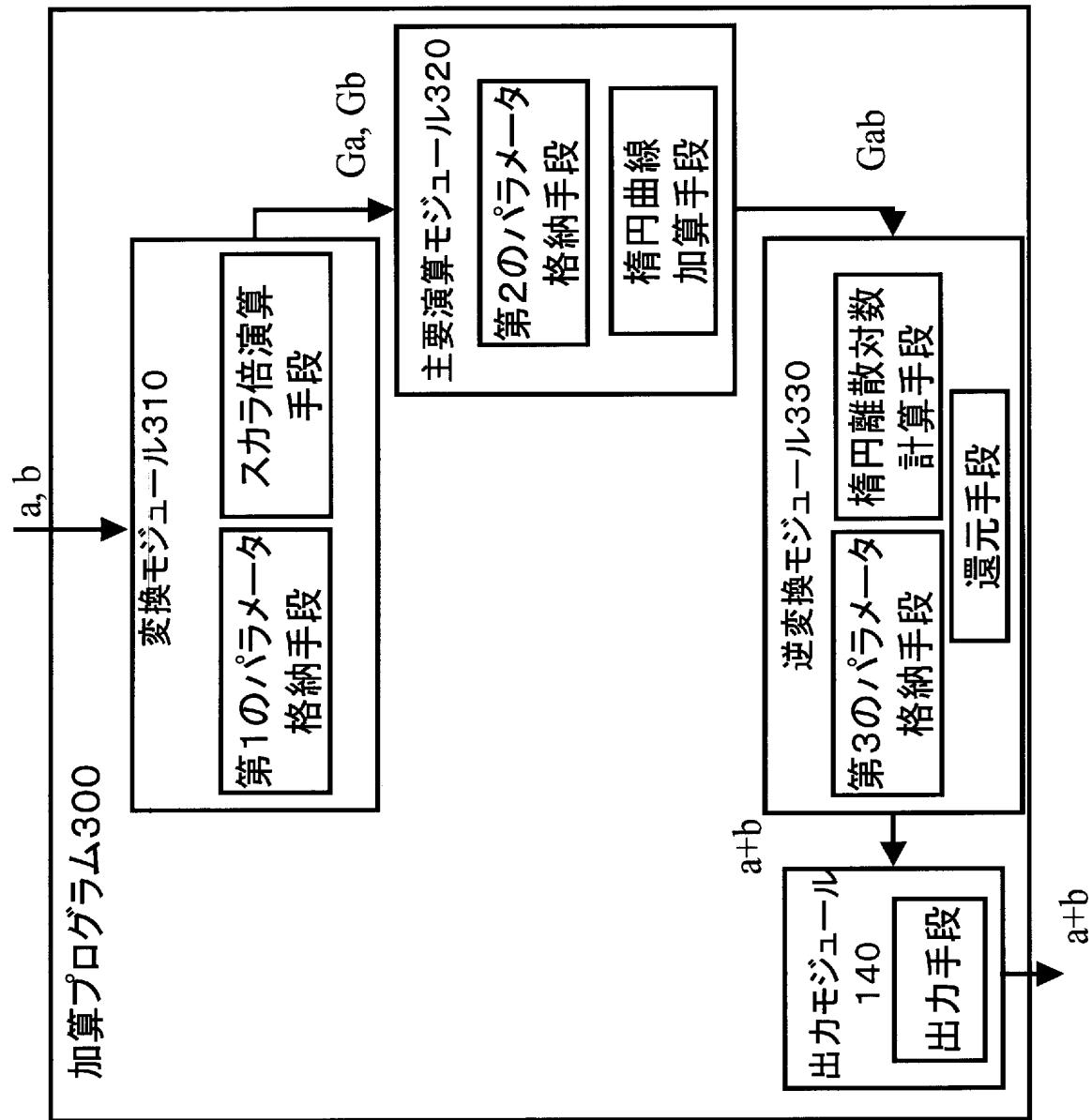
100、200、300、500	加算プログラム
110、210、310、510	変換モジュール
120、220、320、520	主要演算モジュール
130、230、330、530	逆変換モジュール
140、490、540	出力モジュール
400	暗号化プログラム
410	拡張鍵生成モジュール
420	鍵加算モジュール
430	ローテーションモジュールA
440	ローテーションモジュールB
450	ローテーションモジュールC
460	ローテーションモジュールD
470	排他的論理和モジュール
480	ビット分割モジュール



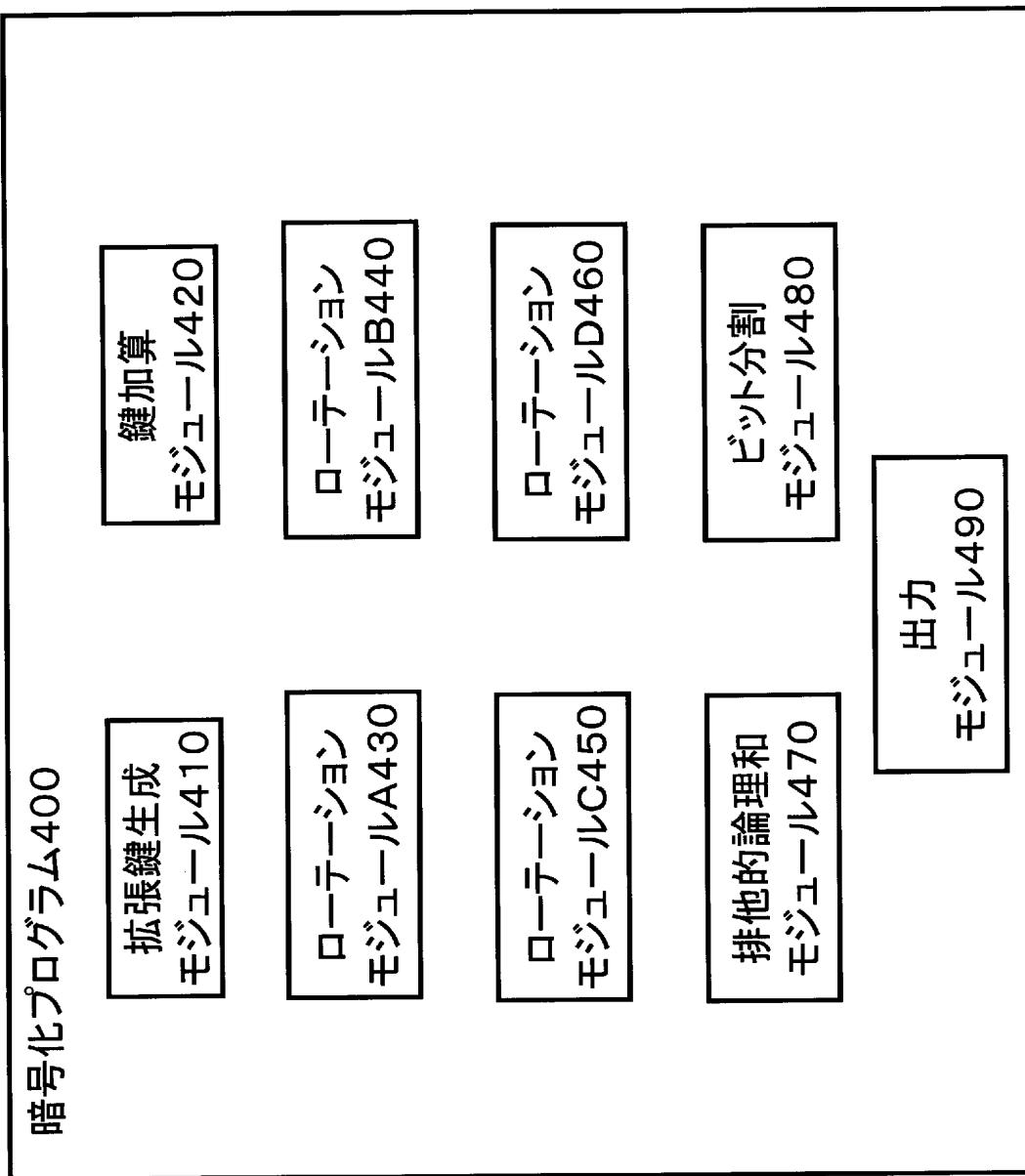
【図 2】



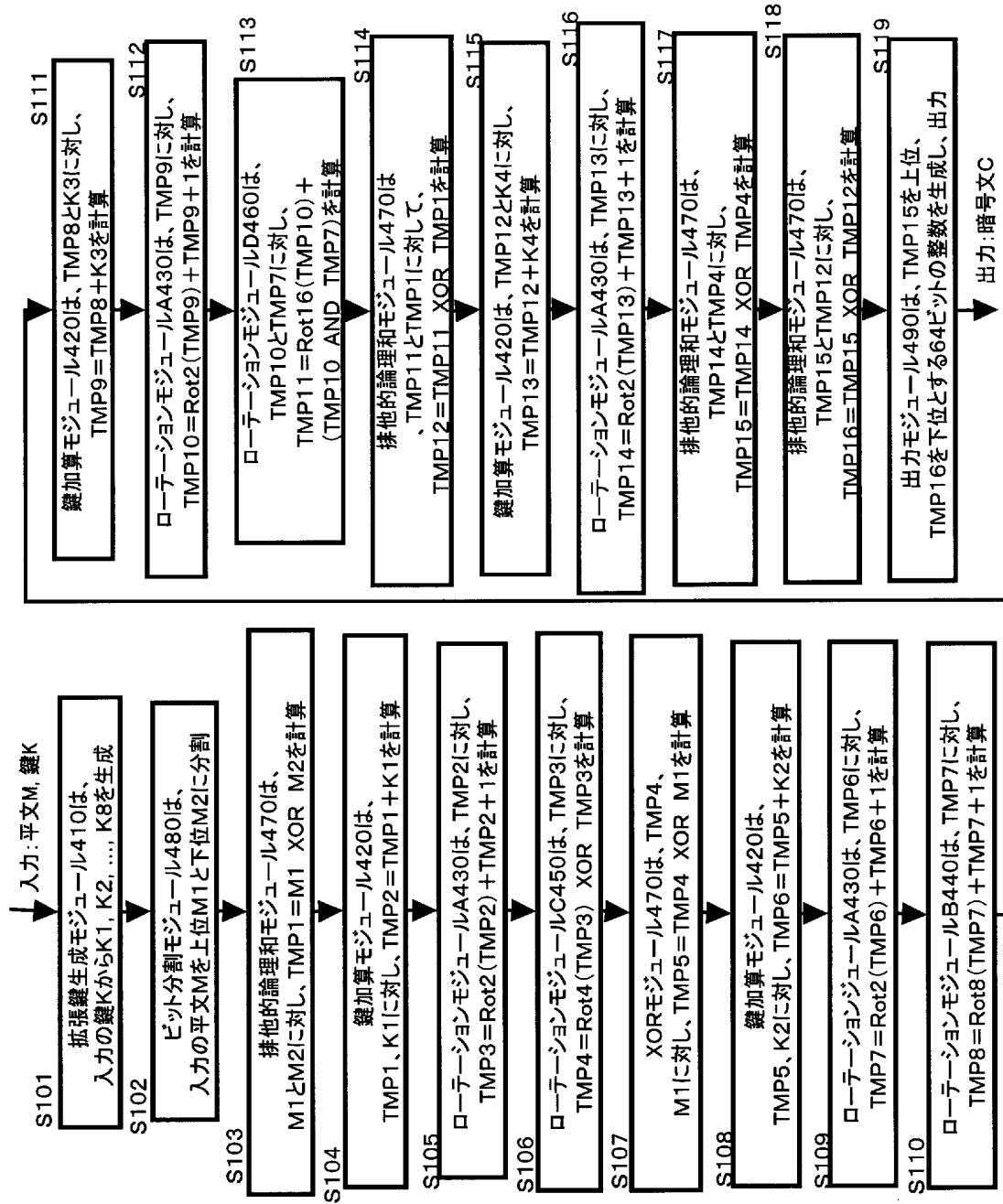
【図 3】



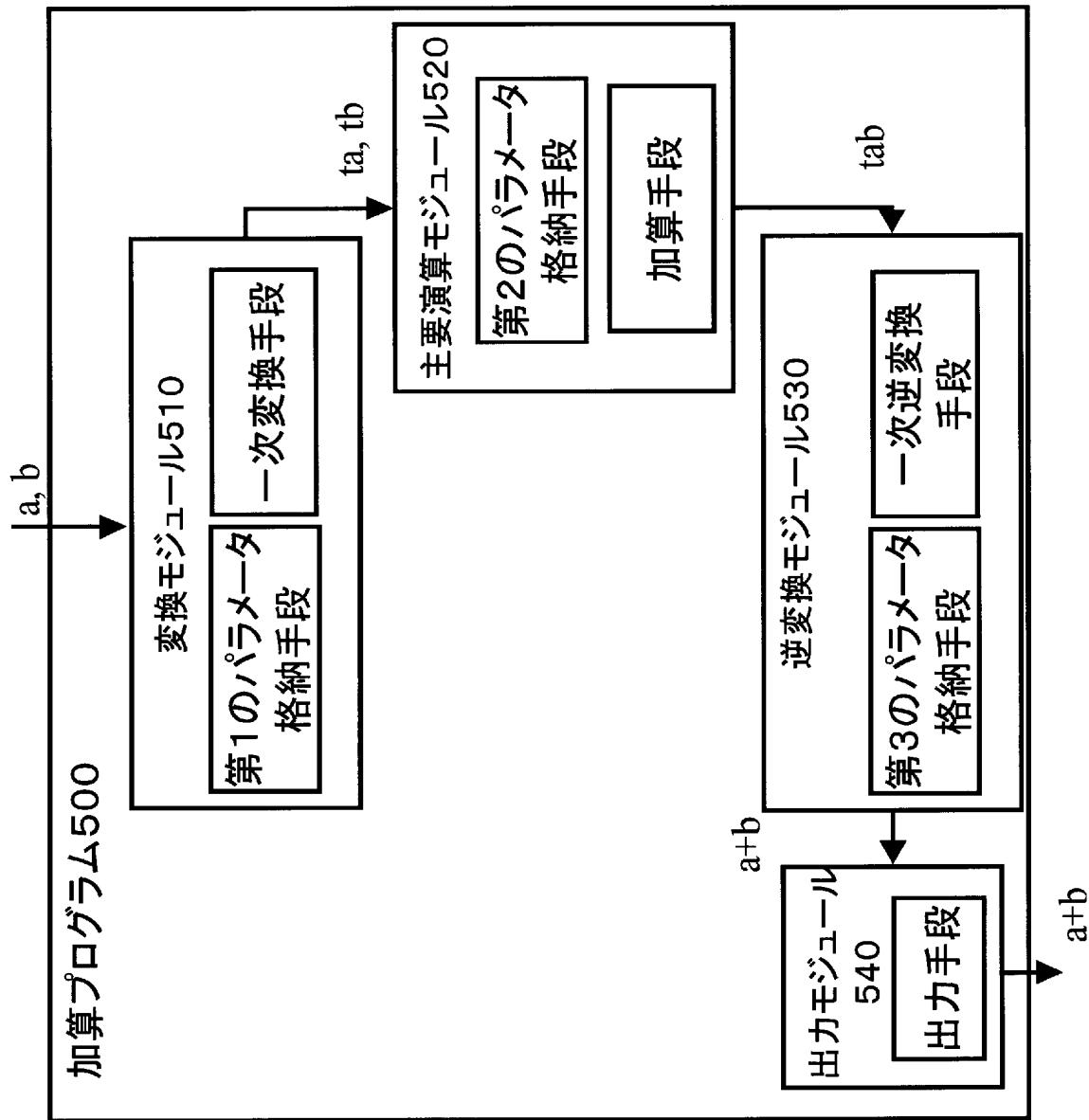
【図 4】



【図 5】



【図 6】



【書類名】要約書

【要約】

【課題】従来の耐タンバーソフト技術として知られている演算の領域を変換する方式では、領域を変換することで値を隠蔽することができるが、何の演算を行っているかは解析者に分かるため、演算の種類に特徴がある場合はその部分を集中して解析される可能性があり望ましくない。そこで、演算そのものも変換する方式が望まれている。

【解決手段】演算を整数加算とした場合に、被演算整数を冪乗演算を用いて剰余整数環の元に変換し、剰余整数環の乗算を行い、その結果を逆変換することにより加算を実現する。

【選択図】 図 2

出願人履歴

0 0 0 0 0 5 8 2 1

19900828

新規登録

大阪府門真市大字門真 1 0 0 6 番地

松下電器産業株式会社